1. Perform arithmetic operation using lamba function

**interface** find

{

**void** action(**int** num1,**int** num2);

}

**public** **class** lambdacheck {

**public** **static** **void** main(String[] args) {

find fn = (num1,num2)->{System.***out***.println("Addition of two number is-> " +(num1+num2) +"\n"+"Substract is-> "+(num1-num2)+"\n"+

"Multiplication is-> "+ (num1\*num2)+"\n"+"Division is -> "+(num1/num2));};

fn.action(12,7);

}

}

1. Print order if order\_price>10000 and order\_status==”ACCEPTED || COMPLETED”

**interface** order

{

**void** find\_order(**int** order\_price,String order\_status);

}

**public** **class** lambdacheck {

**public** **static** **void** main(String[] args) {

order o = (order\_price,order\_status)->{

**if**((order\_price>10000) && (order\_status=="ACCEPTED")|| order\_status=="COMPLETED") {

System.***out***.println(order\_price +" "+order\_status );

}

};

o.find\_order(100000, "COMPLETED");

}

}

1. Use supplier. Functional , predicate,consumer interface methods…

**import** java.util.function.\*;

//consumerclass

**class** consumerr

{

**static** **void** prrint(String s)

{

System.***out***.println("Hello "+ s);

}

}

//functionalclass

**class** Functionn{

**static** String show(String message){

**return** "Hello "+message;

}

}

**public** **class** consumerinterface {

**public** **static** **void** main(String[] args) {

//consumer interface built in method accept

Consumer<String> cs = consumerr::*prrint*;

cs.accept("sumit");

//function interface

Function<String, String> fs = Functionn::*show*;

System.***out***.println(fs.apply("Welcome to my this program"));

//predicate interface

Predicate<Integer> pr = a -> (a > 18); // Creating predicate

System.***out***.println(pr.test(22)); // Calling Predicate method

//Supplier interface.....

Supplier<Double> randomValue = () -> Math.*random*();

// Print the random value using get()

System.***out***.println(randomValue.get());

}

}

4.Removing odd number elements for a list….

Collection<String> names = **new** ArrayList<>();

names.add("John");

names.add("Ana");

names.add("Mary");

names.add("Anthony");

names.add("Mark");

System.***out***.println("List before removing odd kength elements \n");

System.***out***.println(names);

System.***out***.println();

System.***out***.println("List after removing odd kength elements\n");

names.removeIf(e -> length()%2!=0);

System.***out***.println(names);

5. Getting the first letter of a String element. Using a interface cosumer

**interface** consumer

{

**void** first\_index(String[] arr);

}

**public** **class** lambdacheck {

**public** **static** **void** main(String[] args) {

String[] check= {"giant","braddy","wilsom","jackkie"};

consumer ca = (arr)->

{

**for**(**int** i=0;i<arr.length;i++)

{

**char** sb=arr[i].charAt(0);

System.***out***.println(sb);

}

};

ca.first\_index(check);}}

6. Convert lower case string into uppercase using replaceall and unary operator…

//using unary operator....

UnaryOperator<String> caseconversion = (String text) -> { **return** text.toUpperCase();} ;

String newText = caseconversion.apply("wanna learn java wait for 4 months");

System.***out***.println(newText);

//by replaceall() method.....

String s1="wanna learn java wait for 4 months";

String s2= s1.toLowerCase();

String s3 = s1.toUpperCase();

//replaces all small letter to capital

String replaceString=s1.replaceAll(s2,s3);

System.***out***.println(replaceString);

8. print number of list using thread..

**import** java.util.ArrayList;

**import** java.util.Iterator;

**import** java.util.List;

**class** javathread **extends** Thread{

**public** **void** run(){

List<Integer> ar1 = **new** ArrayList();

ar1.add(12);

ar1.add(23);

ar1.add(413);

ar1.add(4111);

Iterator itr=ar1.iterator();

**while**(itr.hasNext()){

System.***out***.println(itr.next());

}

}

**public** **static** **void** main(String args[]){

javathread t1=**new** javathread();

t1.start();

}

}